SPECIFICATION:

Page 1, the Applicant requests that the following paragraphs be inserted before the paragraph in lines 28-31 in page 1.

The following examples clarifies the previous definitions. In the first example, the previous definitions will be exemplified for the following boolean calculation expression:
"(X > 3) or (Y=0) and ((A or B) and (C and not D))"

1. Elements:
X, Y, 3, 0, A, B, C
(It must be noted that X and Y are arithmetic elements while A, B, C are logical elements.)
2. Operators:
or, and, not
3. Comparators:
>, =
4. Items
X>3, Y=0, A, B, C, D
(It must be noted that because this is a boolean expression, X does not have a meaning in itself, because it does not have a boolean value)
5. Subexpressions
Exhibit 1b shows the different types of subexpressions that can be found in the previous example expression.

Exhibit 1b.

Regular subexpresions (because they are items, they are not intermediate subexpressions)	X>3, Y=0, A, B, C, D,

Also, nodes that represent introduced intermediate subexpressions could be conceptualized as regular intermediate subexpressions to which an operator has been added. That is node 9 could be interpreted as been "(C and not D)", and the operator could be added afterwards. Therefore, the key point in order to describe the tree decomposition that is described in this patent application is that at least one of the nodes is an intermediate subexpression, and the subexpression can be understood as either encompassing the introducing operator or not.

Exhibit 13b

No.	Node
0	(X > 3) or (Y=0) and ((A or B) and (C and not D))
1	X>3
2	or (Y=0) and ((A or B) and (C and not D))
3	(Y=0)
4	and (A or B) and (C and not D))
5	A or B
6	A
7	or B
8	and (C and not D)
9	C
10	and not D

Exhibit 13c shows a tree for the same expression that is shown in Exhibit 13b. In this case, node 2 is shown collapsed. It can be seen in this Exhibit that the essence of the invention is that at least one of the nodes is an intermediate subexpression.

Exhibit 13c

No.	Node
0	(X > 3) or (Y=0) and ((A or B) and (C and not D))
1	X>3

2	or (Y=0) and ((A or B) and (C and not D))

Exhibit 13d shows the nature of the invention in more details. In this case, the Exhibit shows a tree decomposition of the same expression that was shown in the two previous Exhibits, but in this case the approach described in US Patent 5,471,613 is utilized. It can be seen that the main difference between the current patent application and US Patent 5,471,613 is that the current patent application shows intermediate subexpressions in the tree, which greatly facilitates comprehending the structure of tree and of the expression.

Exhibit 13d

No.	Node
0	(X > 3) or (Y=0) and ((A or B) and (C and not D))
1	or
2	X>3
3	and
4	Y=0
5	and
6	or
7	A
8	B
9	and not
10	C
11	D

A convenient way to summarize the previous exposition is the following. The tree allows to see at least one fragment of the expression in at least two different nodes, one of the nodes being the parent of the other one. For example, as can be seen in Exhibit 13b, the fragment "A or B" is shown in nodes 0, 2, 4 and 5. In

this case, node 0 is the parent of node 2; node 2 is the parent of node 4, and node 4 is the parent of node 5.

This feature is a crucial part of the invention, because it allows the user to navigate into the expression, allowing him/her to better understand its structure.

This feature is not included in US Patent 5,471,613, and it is not included either in US Patent 6,263,328.

US 6,263,328 discloses an invention to facilitate creating database queries. However, in this invention, there are no intermediate subexpressions. An even though the invention contains a type of representation that might resemble an intermediate subexpression, it does not have any feature to allow the user to access to any part of the expression in two different nodes of the tree. As has been explained, this access is a key part of the current invention.